

활 동 보 고 서

회차	5회차	지도교수 확인란	
일시	2019.8.21.(수) 16:00~19:00		
장소	충남대학교 도서관 그룹 스터디룸		
주제	Java(1) 자바, 이클립스의 설치와 데이터타입, 형변환		
학습 내용 요약			
<div><JAVA 설치와 그 기초></div> <p>Src는 sourc의 약자. 확장자가 .java로 끝나는 파일들이 저장됨</p> <p>Bin은 binary의 약자. 확장자가 .class로 끝나는 파일들이 저장됨</p> <p>우리가 만든 .java로 끝나는 코드파일은 컴파일 과정을 거치면서 .class파일이 된다. 이 파일을 가상머신에 전달해 실행하게 된다.</p> <div><데이터타입>:숫자와문자</div> <p>자바에서는 따옴표가 없는 숫자는 숫자로 인식하며, 문자와 문자열은 다르다,</p> <p>문자(Character)와 문자열(String). 자바에서는 문자는 작은따옴표로 감싸야하고, 문자열은 큰따옴표로 감싸야함. 문자와 문자도 연산자로 더하거나 할 수 있다.</p> <p>큰따옴표를 문자로 사용하고 싶을때는 앞에 역슬래쉬(w)를 붙여서 사용할 수 있다.(이스케이프)</p> <p>줄바꿈은 \n 을 이용한다.</p> <p>Int a; 라고 변수를 선언하는데 여기서 int는 데이터형식이고 a는 변수의 이름이다. Int는 숫자데이터 중 정수타입이다. 그리고 a에 숫자1을 대입(할당)한다. Double은 실수타입이다.</p> <p>실수는 소수점이 있는 수</p> <p>자바는 정수와 실수를 구분해서 사용한다.</p> <p>데이터타입에 따라 메모리용량에 차이가 난다. 따라서 표현하고자하는 변수에 따라 적당한 데이터타입을 선택해야 메모리를 낭비하지않을 수 있다.</p> <div><숫자와 문자></div> <p>자바스크립트에서는 큰따옴표나 작은따옴표가 붙지 않은 숫자는 숫자로 인식해버린다.</p> <p>alert(연산);을 하면 그 결과값을 받을 수 있다.</p> <p>더하기는 + 빼기는 - 곱하기는 * 나누기는 / C언어와 동일함</p>			

여러 가지 연산을 하고 싶을 때는 Math를 쓰고 뒤에는 그 연산의 종류에 해당되는 명령을 쓰면 됨.
pow는 제곱 round는 반올림 ceil는 올림 floor는 내림 sqrt는 제곱근 구하기 random은 난수를 구하는 명령.

문자는 큰따옴표나 작은따옴표로 감싸줘야 문자라고 인식. 물론 큰따옴표는 큰따옴표로 닫고 작은따옴표는 작은따옴표로 닫아야 함. 만약 따옴표를 문자 안에 넣고 싶다면 세 언어와 똑같이 따옴표 앞에 \를 붙여주면 됨.

줄 바꿈을 하고 싶으면 \n를 사용. 문자랑 문자를 더해주려면 alert("문자"+"문자"); 형식으로 사용한다.

ex) alert("I like "+"TEAMLOG");를 입력하면 I like TEAMLOG가 나옴

<변수>

C언어에서는 int라고 변수 선언을 해주지만 자바스크립트에서는 var로 시작한다.

var (변수 이름) = (값); 과 같은 형식. 변수의 이름은 \$, _ 특수 문자를 제외한 모든 문자로 시작할 수 있음. 변수가 없다면 재활용을 하지 못해서 만약 계산할 값을 수정해야 한다면 모든 부분을 바꿔줘야 하기 때문에 변수를 써야 함.

<주석>

주석 처리는 //로. 줄바꿈 전까지만 주석 처리를 해줌.

여러 줄을 주석 처리하고 싶을 때는 /*로 열고 */로 닫아준다 그 사이에 있는 내용들이 주석 처리된다. C언어와 동일.

/** */ "여러줄 짜리 주석이 되면서 JAVADoc주석이라는 문서로 문서화가 되기도 함.

<줄바꿈과 여백>

;는 명령이 끝났다는 것을 명시적으로 표현한 것

자바스크립트는 줄이 바뀌면 명령이 끝났다고 간주. 하지만 명확하게 독립된 명령이라는 것을 알려주기 위해서라면 ;를 사용해야. 탭 키를 눌러주는 이유는 가독성을 위해서.

<비교 연산자>

> , < , <= , >= , =

==는 좌항과 우항의 값이 같다면 true 다르다면 false가 됨.

!=는 부정을 의미한다 같지 않다는 뜻은 !=로 표시한다.

!==도 있는데 != 과 ==의 관계랑 같다.

학습 방법 및 과정

4회차부터 그랬듯, 스터디룸을 빌릴 수 있는 시간이 길지 않기 때문에 각자 학습을 한 후 이해가 안가는 부분과 인상깊었던 부분이나 이번 주차에서 기억해야 할 부분을 이야기 해보는 방식으로 스터디를 진행하고 있다. 다음은 학우들이 각자 인상깊었다고 한 부분이다.

신예슬 학우: 문자를 큰따옴표로 감싸다고 에러가 나진 않지만, 문자열을 작은따옴표로 감싸면 에러가 발생한다. 파이썬과 차이점이 있음.

강미규 학우: 생활코딩 강좌 내에서 나왔던 깨알 팁. 유용할 것 같음. 개발자 도구를 사용해야 오류가 있는지 없는지 찾을 수 있다. 그 오른쪽에 파일명은 그 파일이 에러가 발생하고 있는 것이고 그 옆에 있는 숫자는 그 숫자 줄이 문제라는 뜻. Teamlog.html:8 이면 Teamlog.html 파일에 8번째 줄이 문제가 있다는 뜻.

김예린 학우: 설치 경로 bin폴더에 자바형식으로 코딩한 코드를 저장한 후 그것을 cmd 입력창에 경로를 입력했다. 그 후 경로를 포함한 줄바꿈 화면에 파일명을 입력했더니 별 반응이 없었는데, dir/w 이란 명령어를 입력하자 해당 폴더에 HelloWorld라는 해당폴더의 클래스파일이 생긴 것을 확인할 수 있었다. 이것은 자바에 포함된 버추얼머신이 인식할 수 있는 파일으로, 실행이 가능해졌음
(김예린 학우의 경우에는 강의의 내용이 아닌, 코딩을 하다가 에러가 나 학우들과 함께 디버깅을 한 것을 토대로 작성하였다)

이정란 학우가 배운 것 중에서 중요하게 여겨진다고 한 비교 연산자에 대해 보충설명을 해 주었다.
" ===라는 것이 있는데 예를 들어

`alert(1=='1');`는 true

`alert(1==='1');`는 false

처음 볼 때는 이상하지만 ===연산자는 같은 수여도 데이터 형이 같아야만 같다고 판단하기 때문에 false라는 결과가 뜬 것이다.

null은 값이 없다는 뜻이고 undefined는 값이 정의되지 않았다는 뜻이다. 똑같이 값이 없는 상태지만 undefined는 프로그래머가 의도하지 않은 상황이다.

NULL은 프로그래머가 의도적으로 현재 값이 없다는 것을 지정한 상황이다.

NaN은 특수한 데이터 형인데 숫자가 아니라는 뜻이다. "

김현지 학우: 자바와 같은 언어는 효율적인 프로그래밍을 위해 숫자의 구분을 하며, 그 종류에는 정수형, 실수형, 문자형 등이 있음.

<학습 소감>

신예슬 학우: 파이썬을 먼저 배워서 그런지 배우면서 계속 파이썬과 비교를 하게 됐다. 일단 자바와 자바의 대표적인 개발도구인 이클립스는 설치부터가 참 어려웠다. 파이썬에 비해 많이 헤맸던 것 같다. 그래서 설치와 실행하는 데까지 꽤 오랜시간이 걸렸고 팀원들끼리 많이 의논하고 도와줬던 것 같다. 그래서 첫 인상부터 자바는 참 만만치 않은 언어라는 걸 깨달았다. 파이썬에 비해 여러 가지 신경 써야 할 부분들이 많았지만, 그만큼 배울점이 많았다. 세세한 규칙들에는 다 합당한 이유가 있었고 합리적이었기 때문이다.

강미규 학우: 그리고 초반부분을 들으면서 파이썬이 왜 입문자들이 처음 배우면 좋은 언어인지 더욱 알게 되었다. 자바는 파이썬보다 훨씬 간간하고 좋게말하면 더 확실하고 간간한 만큼 오류가 없는 체계적인 언어라는 느낌이 들었다. 한 가지 난관이 있었다면 모든 줄의 끝에 ;를 붙이는 문법인데, 파이썬이나 루비에서는 없는 규칙이고, 너무 작은 오류여서 오류가 생기면 왜 오류가 생겼는지 몰라 한참을 다들 코드를 보고 또 보곤 했다. 어렵지만 그래도 더 열심히 해보려 한다.

이정란 학우: 크게 보면 C언어랑 비슷한 부분이 많아서 쉽게 들을 수 있었지만 객체 지향 같은 경우에는 이해하기 힘들어서 같은 동아리 내 회원들의 도움을 받게 되었다.

또한 나는 개인적으로 코딩이 회사 실무에서 얼마나 많이 쓰이는지 찾아봤는데 수업이 실무와 약간의 차이가 있을 수는 있다고 했다. 회사업체에서는 이클립스 뿐만 아니라 다른 툴을 사용할 수 있기 때문이다. (실제로 한 가지만 사용하는 것은 아니니까)

하지만 강의에서 배우는 코딩의 기본개념을 파악한다면 충분히 실무에서도 차이가 있더라도 실무에 대해서 이해하고 응용해 나갈 수 있는 능력이 훨씬 편할 것이라고 생각한다. 수업 잘 듣고, 최대한 수업을 들으면서 이해하고, 구조를 파악하는데 중점을 두고, 가끔씩은 다른 곳에서는 어떤 식으로 응용이 될 수 있을까 관심을 가지면 충분히 강의를 잘 활용할 수 있을 것이라고 확신한다!

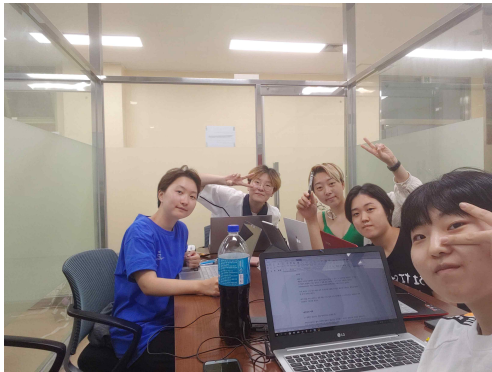
김예린 학우: 자바는 이전에 배웠던 파이썬, 루비와 다르게 데이터 타입을 구분하는 것이 가장 큰 특징처럼 느껴졌다. 예를 들어서 200이라는 것을 표현하려면 short 혹은 int라는 것을 써야 하는 것처럼, 엄격하지만 무엇이 어디에 들어가야 할지 정해져 있는 정갈하고 단정한 언어인 것 같다는 인상이 들었다. 또한 파이썬에 비해서 더 오류가 잘 나고, 대소문자의 구분이 엄격한 느낌이 들었는데, 다른 회원들과 이야기를 해 본 결과 다른 회원들도 나와 비슷한 생각을 한 것 같다. 또 강의와는 별개지만, 나는 자바를 파이썬에 비해 그 중요성을 잘 못 느꼈는데, 오히려 앱 개발이나 게임 개발을 하려면 자바가 더 필요하지 않을까 하는 생각이 들어 더 열심히 배워야겠다는 생각을 했다.

김현지 학우: 자바가 규모가 큰 소프트웨어를 만드는 프로그램 언어라는 것을 처음 알았다. 일단 소개를 읽어봤을 때 왠지 노트북에서는 렉이 많이 걸릴 것 같았고, 실제로 이클립스까지 설치하는 과정에서 노트북이 너무 느려져서 걱정했다. 그래도 설치했다. 그냥 실행하면 되는 파이썬이나 루비와 달리, 자바는 cmd에서 실행하는 데 컴파일을 해야한다는 점이 신기했다. 컴파일을 하는 것이 처음이어서 조금 헤맸고, 하라는 대로 바르게 했더니 이번엔 실행하려는 자바 버전이 컴파일하는 자바 버전과 달라서 에러가 났다. 이클립스를 실행하니 모니터에 여러 가지 동작을 할 수 있는 뷰가 많았다. 더 무겁고 복잡하고 어떻게 보면 더 전문적인 것처럼 보이는 프로그래밍 언어라는 게 자바에 대한 나의 첫인상이었다. 그 뒤 강의에서 배운, 변수를 선언하지 않으면 사용할 수 없다는 점이나, 데이터 값에 따라 데이터 타입을 다르게 써야 하거나 형 변환이라는 것에 대해 알아야 한다는 점은 자바라

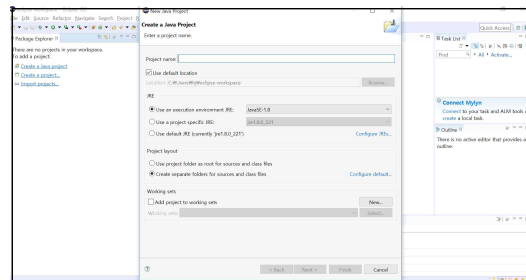
확실히 체계적이라는 느낌을 주었다.

활동 사진 첨부

학습 활동 ①



학습 결과물 ①



이클립스에서 자바를 실행하는 장면
(김현지 학우)

학습 활동 ②



(김예린 학우의 원손)

학습 결과물 ②

```
1 package org.opentutorials.javatutorials.Variables;
2
3 public class IntegerVariables {
4
5     public static void main(String[] args) {
6         //int a;
7         a = 1;
8         System.out.println(a+1); //2
9
10        a = 2;
11        System.out.println(a+1); //3
12    }
13 }
14
15 }
16
```

김현지 학우의 이클립스 화면
: int 변수 선언의 예시

